HYBRIDIZING MULTI-INVER-OVER EVOLUTIONARY ALGORITHMS WITH TABU SEARCH FOR THE SYMMETRIC TSP

Bermudez C., Minetti G., Alfonso H.
Laboratorio de Investigación en Sistemas Inteligentes (LISI)[1]
Departamento de Informática - Facultad de Ingeniería
Universidad Nacional de La Pampa
Calle 110 esq. 9
(6360) General Pico – La Pampa – Rep. Argentina
e-mail: {bermudec, minettig, alfonsoh }@ing.unlpam.edu.ar
Phone: (02302) 422780/422372, Ext. 6302

Gallard R.
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)[2]
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950 - Local 106
5700 - San Luis- Rep. Argentina
e-mail: rgallard@unsl.edu.ar

ABSTRACT

The travelling salesman problem (TSP) is a NP-hard problem. Techniques as either Branch and Bound or Dynamic Programming supplied the global optimum solution for instances with more than 7000 cities. But, they needed more than 4 years of CPU time. Fortunately, faster algorithms (simulated annealing, tabu search, neural networks, and evolutionary computation) exist although they do not guarantee to find the global optimum.

Recently an EA based on a operator *inver-over* [4], provides optimal or near-optimal solutions in a very short time. A latest approach included a variant of inver-over called multi-inver-over [6]. The corresponding results showed advances when compared with other search techniques.

This work shows a further enhancement, the Hybrid Multi-inver-over Evolutionary Algorithms (HMEAs), which consists in hybridizing multirecombined evolutionary algorithms with Tabu Search. In these algorithms local search is inserted in different stages of the evolutionary process as in [7 and 8]. They were tested on the hardest set of the test suite chosen in previous works. Details on implementation, experiments and results are discussed.

Keywords: TSP, hybridization, multirecombination, tabu search, evolutionary algorithm

---

[1] The Research Group is supported by the Universidad Nacional de La Pampa.
[2] The LIDIC is supported by the Universidad Nacional de San Luis and the ANPCYT (National Agency to Promote Science and Technology).

1. PROBLEM DESCRIPTION

TSP is one of the most widely studied and challenging NP-hard combinatorial optimization problems. Here, a travelling salesman wants to visit each of n cities starting and ending at a designated city 1. He visits no other city twice. Let $c_{ij} \geq 0$ the cost (or distance) between city i and city j. When there is no direct connection between them, we assign $c_{ij} = \infty$. The optimization problem is to find a minimum cost (shortest) tour. There are several mathematical formulations of the problem. The following one uses relatively few variables. Define the zero-one variables:

$$x_{ij} = \begin{cases} 1 \text{ if a tour includes travelling from city } i \text{ to city } j \\ 0 \text{ otherwise} \end{cases}$$

for all i and j.
The objective is to minimize,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \quad \text{where } c_{ii} = \infty \text{ for } i = 1,..,n$$

s.t.

$$. \sum_{j=1}^{n} x_{ij} = 1 \quad \text{for } i = 1,..,n \text{ (departure)} \qquad (1)$$

$$. \sum_{i=1}^{n} x_{ij} = 1 \quad \text{for } j = 1,..,n \text{ (arrival)} \qquad (2)$$

$$x_{ij} \text{ nonnegative integers for all } i \text{ and } j. \qquad (3)$$

Restrictions (1), (2) and (3) ensure that each x_{ij} is either zero or one. Restriction (1) requires that a tour include one departure from each city, while (2) guarantees one arrival at each city.

2. THE EVOLUTIONARY ALGORITHM WITH INVER-OVER OPERATOR AND MULTIRECOMBINATION

This algorithm [4] works with a population of a single individual, where each parent can be replaced by its unique child, or vice versa. Unlike traditional EAs, which use two genetic operators (i.e, mutation and crossover), this new evolutionary algorithm uses only the inver-over operator which combine the mutation and recombination. The mutation is done through inversions in a segment of the same individuals; but the segment to be inverted depends on the other individual in the population (recombination).

The evolutionary algorithm based on the inver-over operator can be seen as a set of m-parallel hill climbing procedures. The method provided optimal or near-optimal solution very fast and outperformed other evolutionary operators proposed in the past for the TSP like PMX, OX, CX, ER, etc.

Taking that idea we tried to improve it by incorporating multiplicity features [1, 2]. Expecting to find better solutions, the inver-over operator is applied a number n_I of times on the current individual. In this way the multi-inver-over operator is created and many other individuals can offer their knowledge to create a better solution. So, when the evaluations of the new and the original individuals are compared, if the new one does not improve the original then the loop for the inver-over operation is repeated, for a maximum number n_I of times. As a subclass of MFEAs (Multiplicity Features Evolutonary Algorithms), in previous works these algorithms were called *multiple inver-over evolutionary algorithms* (MEAs) and according to the number of operations to be applied to a single solution they were identified as IO-n_I (standing for "Inver Over n_I times").

3. TABU SEARCH

The main idea behind tabu search is related with a memory, which drives the exploration toward new regions of the problem space. Some solutions, which have been tested recently in the past, can be memorised and became tabu (forbidden) points. In this way, these solutions are not tested again during some predetermined time interval h (horizon). As a consequence, Tabu Search is essentially deterministic.

Here, a particular Tabu Search procedure is used for the TSP [5] and it is explained through the following example. A solution with n = 8 cities is considered and the neighbours are produced by swapping two cities in a particular solution. Let (2, 4, 7, 5, 1, 8, 3, 6) be the initial solution. Hence $\binom{n}{2} = \frac{n(n-1)}{2} = 28$ neighbours are possible.

Two different memories are used; a *recency-based memory* and a *frequency-based memory.* The first indicates the number of remaining iterations for which a given swap stays on the tabu list, while the second indicates the total number of swaps that occurred within some horizon h. For a recency-based memory the structure given in figure 1 can be used, where the interchange of cities i and j is recorded in the i-th row and j-th column (for $i < j$). The same structure can also be used for frequency-based memory.

2	3	4	5	6	7	8	
							1
■							2
	■						3
		■					4
			■				5
				■			6
					■		7

Figure 1: the structure of the recency-based memory for the TSP

Assume both memories were initialized to zero and 500 iterations of the search process have been completed. The current status of the search then might be as follows.

The current solution is (7, 3, 5, 6, 1, 2, 4, 8) with the total length of the tour being 173. The best solution encountered during these 500 iterations yields a value of 171. The status of the recency-based and frequency-based memories are displayed in figures 2 and 3, respectively.

2	3	4	5	6	7	8	
0	0	1	0	0	0	0	1
■	0	0	0	5	0	0	2
	■	0	0	0	4	0	3
		■	3	0	0	0	4
			■	0	0	2	5
				■	0	0	6
					■	0	7

Figure 2: the contents of the recency-based memory M for the TSP after 500 iterations. The horizon is 5 iterations.

The value M (2, 6) = 5 indicates that the most recent swap was made for cities 2 and 6, i. e., the previous current solution was (7, 3, 5, 2, 1, 6, 4, 8).
Therefore, swapping cities 2 and 6 is *tabu* for the next five iterations. Similarly, swaps of cities 1 and 4, 3 and 7, 4 and 5, and 5 and 8, are also on the tabu list. Among them, the swap between cities 1 and 4 is the oldest (it happened five iterations ago) and this swap will be removed from the tabu list after the next iteration. Note that only 5 (recency-based horizon) swaps (out of 28 possible swaps) are forbidden (tabu).

2	3	4	5	6	7	8	
0	2	3	3	0	1	1	1
■	2	1	2	1	1	0	2
	■	2	2	3	4	0	3
		■	1	1	2	1	4
			■	4	2	1	5
				■	3	1	6
					■	6	7

Figure 3: the contents of the frequency-based memory F for the TSP after 500 iterations. The horizon is 50 iterations.

The frequency-based memory provides some additional statistics of the search. It indicates that swapping cities 7 and 8 was the most frequent move (it happened 6 times in the last 50 –frequency-recency horizon– swaps), and there were pairs of cities (e.g. 3 and 8) that weren't swapped within the last 50 iterations. This kind of memory might useful to diversify the search. Because it provides information concerning which flips have been under-represented (i.e less frequent) or not represented at all, and then diversify the search by exploring these possibilities.
There are many possibilities here for incorporating this information into the decision-making process. The most typical approach makes the most frequent moves less attractive. Usually the value of the evaluation score is decreased by some penalty measure that depends on the frequency, and the final score indicates the winner.
The following figure shows an implementation of tabu search reported in [3].

```
procedure tabu search
begin
  tries = 0
  repeat
    generate a tour
    count = 0
    repeat
        identify a set T of 2-interchange moves
        select the best admissible move from T
        make appropriate 2-interchange
        update tabu list and other variables
        if the new tour is the best-so-far for a given tries
        then update local best tour information
        count = count +1
    until count = ITER
    tries = tries +1
    if the current tour is the best-so-far (for all tries)
    then update global best tour information
  until tries = MAX-TRIES
end
```

Fig. 4. Implementation of tabu search for the TSP.

4. HYBRID ALGORITHMS

The Hybrid Multi-inver-over Evolutionary Algorithms (HMEAs) presented here incorporate TS to the IO-n_I versions, in distinct stages of the evolutionary process, as follows:

- HMEA-IP. Applies TS to each individual of the initial population and the EA begins from this improved population.
- HMEA-MP. Applies TS to some individuals of intermediate populations. The decision to apply TS is taken according to the following policy. After a given number φ of generations, a control of the convergence begins. From that point in the evolutionary process, when after a given number λ of consecutive generations no improvement is detected in the best solution, then TS is applied. This approach has the following variants,
  - ✓ HMEA-MPB10 and HMEA-MPB20. They apply TS to the 10% or 20%, respectively, of the best individuals in the population.
  - ✓ HMEA-MPW10 and HMEA-MPW20. They apply TS to the 10% or 20%, respectively, of the worst individuals in the population
  - ✓ HMEA-MPR10 and HMEA-MPR20. They apply TS to the 10% or 20%, respectively, of randomly selected individuals in the population.
- HMEA-FP. Applies TS to individuals of the final population. Similarly to HMEA-MP, this approach has the following three variants.
  - ✓ HMEA-FPB10 and HMEA-FPB20. They apply TS to the 10% or 20%, respectively, of the best individuals in the final population.
  - ✓ HMEA-FPW10 and HMEA-FPW20. They apply TS to the 10% or 20%, respectively, of the worst individuals in the final population
  - ✓ HMEA-FPR10 and HMEA-FPR20. They apply TS to the 10% or 20%, respectively, of randomly selected individuals in the final population.

5. EXPERIMENTS

According to the above described hybrid algorithms, a set of experiments was performed. All of them used the multirecombinative inver-over operator. Five different approaches, *IO-1* to *IO-5,* were conducted applying from 1 to 5 inver-over operations, respectively.
All approaches were tested for five TSP instances, extracted from TSPLIB95 [9]. They were:

Identifier	Instance	Best know Value
B127	Bier127 (127 cities)	118282
E101	Eil101 (101 cities)	629
Kc100	Kroc100 (100 cities)	20749
P76	Pr76 (76 cities)	108159
S70	St70 (70 cities)	675

For each instance a series of ten runs was performed. All the EAs used the following parameter settings:

Population size	100
Probability p	0.02
Stop criterion	After the 500th generation, if the best individual does not change during 100 consecutive generations.
Maximum No. of generations	4000
Elitism	Yes

As the number of cities is considerable, a different way to obtain the neighbourhood was necessary to devise. For each solution, where TS was applied, a random city was selected; after that the neighbours were created interchanging this city with every other city. In this way n-1 neighbours are considered. To determine when TS should be applied φ and λ were fixed in 200 and 50 generations, respectively.
Parameter settings for TS were the following:

Maximun No. of iterations	1000
Recency-based horizon	5
Frequency-based horizon	50
Penalty	0.7

The main idea of this implementation is to intensify local search by means of Tabu Search inserted into the EA, while the global search is mainly carried out by the inver-over and multi-inver-over operators. This would allow solving large instances of TSP effectively and efficiently.
To measure the performance of the algorithm the following relevant variables were chosen:

- ✓ **Ebest** = (Abs(*opt_val* - best value)/*opt_val*)100. It is the percentile error of the best found individual when compared with the known, or estimated, optimum value *opt_val*. It gives us a measure of how far the best individual is from that *opt_val*.

- ✓ **Epop** = (Abs(*opt_val*- pop mean fitness)/*opt_val*)100. It is the percentile error of the population mean fitness when compared with *opt_val*. It tell us how far the average individual is from that *opt_val*.

- ✓ **Gbest** : Identifies the generation where the best individual (retained by elitism) was found.

6. RESULTS

Ebest values show how good the different approaches are to find their best solutions. In table 1 and figure 5 an overview (through all recombinative approaches *IO-1* to *IO-5*) with the most representative cases is shown. Regarding the quality of the results from the comparison between the multi-inver-over EA (MEA) and the Hybrid Multi-inver-over EAs (HMEAs), a significant improvement in the performance is observed when the latter are applied. Besides, contrasting the distinct HMEAs, better results are obtained when the hybridization is applied either in intermediate or in final populations instead of in the initial one. Although for E101 instance, this difference is very small.
When HMEA-IP is used 100 applications of TS are done (once on each individual of the population). In HMEA-MPs, this number depends on the amount of generations the approach needs to reach its best individual. HMEA-FPs apply TS only in the 10% or 20% of the population only once. In consequence the latter approach uses much less computational effort than the others. Besides the quality of results is similar or better than those obtained by HMEA-MPs.
Given the great number of combinations between instances and recombinative approaches, only the results for the Kc100 instance are shown in detail in the next section, for each recombinative approach of the hibridized versions. This instance presents a good mixture of number of cities and distance among them.

Instance	MEA	HMEA (IP)	HMEA (MP-R20)	HMEA (FP-R20)	HMEA (MP-B20)	HMEA (FP-B20)	HMEA (MP-W20)	HMEA (FP-W20)
B127	9.531908021	6.643986541	4.897622512	5.013367444	5.206745912	4.392854382	5.143097733	5.388130738
E101	10.73360034	7.850241653	7.068426073	7.370883943	6.981647059	7.337882353	7.233173291	7.498937997
Kc100	13.10586964	7.053017784	4.901198323	5.911047376	4.84217996	4.749716902	5.95811509	5.021373271
P76	10.72419951	3.087622815	2.854817704	2.816946884	2.634709419	2.544089054	2.610719515	3.075676865
S70	8.088560403	5.488480000	2.940568889	3.148094815	3.182912593	3.726850370	3.530717037	3.948506667

Table 1. : Average Mean Ebest for each instance

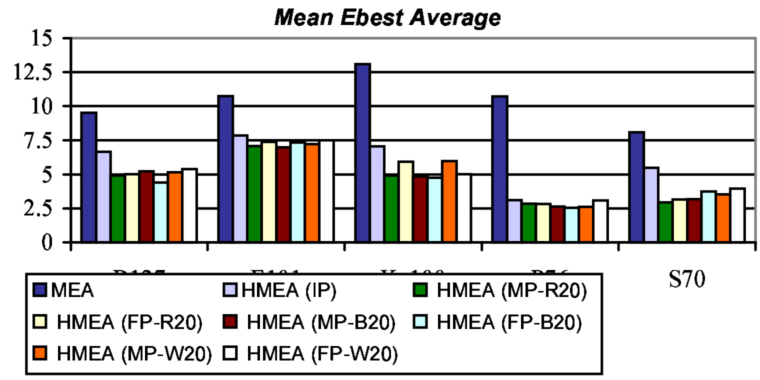

Fig. 5. : Average Mean Ebest for each instance

6.1. KROC100 INSTANCE

From the analysis of the mean *Ebest* values for Kc100 instance (table 2 and figure 6), we can remark that:

- Those HMEAs, where the best individuals were selected, outperform others approaches.
- When worst or random individuals are selected, significant differences are not found.
- All the algorithmic approaches obtain better solutions when the number of recombinations grows. Note the great step generated between IO-1 and IO-2, which gives an idea of how important the multirecombination is.

Method	HMEA (MP-R20)	HMEA (FP-R20)	HMEA (MP-B20)	HMEA (FP-B20)	HMEA (MP-W20)	HMEA (FP-W20)
IO-1	9.75899610	10.44327871	10.1400949	6.853086896	11.13703938	8.940117114
IO-2	4.91490192	6.673065690	4.58555545	5.345102415	6.815142898	5.300949925
IO-3	3.80093691	5.260825582	3.76471300	3.703152923	3.925253747	4.346898164
IO-4	3.28162658	4.006229698	3.00280110	4.224619018	4.234506723	3.607374331
IO-5	2.74953010	3.171837197	2.71773531	3.622623259	3.678632705	2.911526821

Table 2: Mean Ebest for Kc100 instance

Regarding the error of the average individual in the population (*Epop*), a similar behavior of the algorithms can be observed in figure 7. Significant differences between the best individual and the average individual in the population (fig 6 and 7) are detected for each algorithmic approach. This is an indication that final populations are highly diverse and, if desired, the search could be continued to obtain convergence of the population towards the best individual.
From figure 8 we conclude that *Gbest*, the generation where the best individual is found, also decreases as long as n_I increases. This behaviour, at some degree compensates the extra computational effort required for multirecombination.

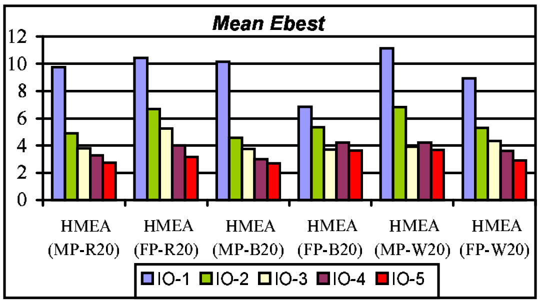

Fig. 6: Mean Ebest for Kc100 instance

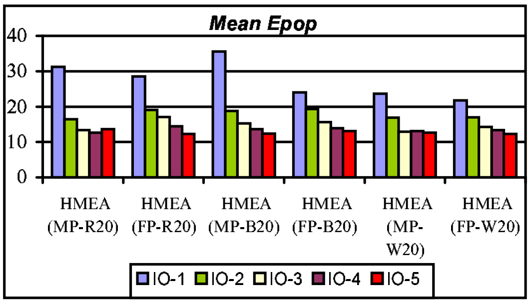

Fig. 7: Mean Epop for Kc100 instance

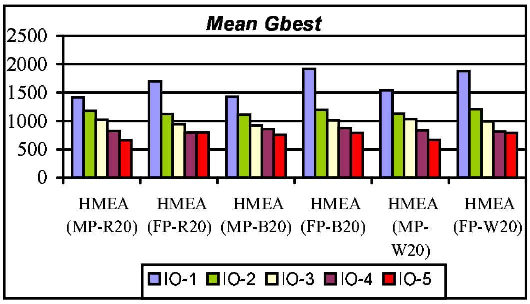

Fig. 8: Mean Gbest for Kc100 instance

7. CONCLUSIONS

In this work a Hybridization of Multi-inver-over Evolutionary Algorithms (HMEAs), by means of Tabu Search (TS), is presented to solve hard TSP instances. The Tabu Search procedure was implemented using two different memories: recency-based and frequency-based. When compared with previously implemented Multi-inver-over Evolutionary Algorithms (MEAs) we see that better solutions for these hard instances of the travelling salesman problem were found under HMEAs.
Three main HMEAs were implemented according to the criterion establishing when hybridization should be applied: HMEA-IP (initial population), HMEA-MP (intermediate populations) and HMEA-FP (final population). Regarding the quality of results analysed through the performance variables, all hybrid approaches improve results obtained by MEAs with diverse extra effort. HMEA-IP allowing the evolutionary process to start with a better initial population provides solutions of significantly lower quality than those obtained by other hybrid variants. HMEA-MP and HMEA-FP provide similar quality of results, but HMEA-FP requires lesser computational effort and consequently is the recommended hybrid option to solve TSP.
All hybrid approaches show substantial improvements as long as the multiplicity of the inver-over operations is incremented. Future work will be devoted to further study this influence on diverse local search approaches to hybridize evolutionary algorithms.

8. ACKNOWLEDGEMENTS

We acknowledge the cooperation of the project groups for providing new ideas and constructive criticisms. Also to the Universidad Nacional de San Luis, the Universidad Nacional de La Pampa, and the ANPCYT from which we receive continuous support.

9. REFERENCES

[1] Esquivel S., Leiva A., Gallard R., - "Multiple Crossover per Couple in Genetic Algorithms", Proceedings of the Fourth IEEE Conference on Evolutionary Computation (ICEC'97), pp 103-106, ISBN 0-7803-3949-5, Indianapolis, USA, April 1997.

[2] Esquivel S., Leiva H.,.Gallard R., "Multiple crossovers between multiple parents to improve search in evolutionary algorithms", Proceedings of the 1999 Congress on Evolutionary Computation (IEEE). Washington DC, pp 1589-1594.

[3] Knox J., Tabu Search Performance on the Symmetric Travelling Salesman Problem. Computer Operations Research, Vol. 21, No. 8, pp. 867-876, 1994.

[4] Michalewicz Z., Esquivel S., Gallard R., Michalewicz M., Tao G, Trojanowski, K., The Spirit of Evolutionary Algorithms, special issue on Evolutionary Computing of the Journal of Computing and Information Technology, University Computing Centre, Zagreb, Croatia, pag.1-18, 1999.

[5] Michalewicz Z, Fogel D.B., How to solve it: modern heuristics, Springer-Verlag Heidelberg, Germany 2000.

[6] Minetti G., Gallard R., Alfonso H., Inver-Over variants for the Euclidean Travelling Salesman Problem, Proceedings of the Second International ICSC Symposium on Engineering Of Intelligent Systems, EIS'2000, Escocia, pag. 458-463, 2000.

[7] Minetti G., Gallard R., Alfonso H., Hybrid Multi-Inver-Over Evolutionary Algorithms for the Euclidean Travelling Salesman Problem, pag.134, Invited Session 'Current trends in Evolutionary Computation to Face Scheduling Problems', SOCO'2001, 2001.

[8] Minetti G., Gallard R., Alfonso H., Influence of temperatures settings for simulated annealing in a hybrid evolutionary algorithm when solving the TSP, proceedings of the SCI 2001, Vol. III, pp. 397-400, 2001.

[9] Reinelt, G., TSPLIB 95, Universität Heidelberg, Institut für Angewandte Mathematik, 1995.